IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
Research Paper **© 2012 IJFANS. All Rights Reserved**, **Journal Volume 11, Iss 07, 2022**

64-BIT FPGA-BASED RANDOM NUMBER GENERATOR

[#1]NARAHARI SUMA, ***Asst. Professor,***

[#2]KOTICHINTHALA NEETHIKA, ***Asst. Professor,***

Department of Electronics Communication Engineering,

Sree Chaitanya Institute Of Technological Sciences, Karimnagar, Ts.

Abstract: A true random number generator (TRNG) is required in a wide range of critical security applications. Despite their frequent reliance on analog randomness sources, digital solutions play a crucial role, particularly in the design of FPGA-based digital systems. This work introduces a novel way for accelerating the development of a TRNG by utilizing Field Programmable Gate Array (FPGA) components. DCM hardware primitives can be used to alter the phase difference between two clock signals at runtime. By individually altering the phase difference between two clock signals, the metastability zone is created in one or more flip-flops (FFs) using the technique described below. In the current system, the aforementioned area is a key source of uncertainty. This paper also proposes a novel implementation of the quick carry-chain hardware primitive to boost the randomness of the generated bits. This paper describes a powerful on-chip post-processing solution for True Random Number Generators (TRNGs) that prevents output slowing. Throughout development, Verilog Hardware Description Language (HDL) was used, with 32- and 64-bit data widths. In order to finish the synthesis, QUARTUS II was used. Throughout the evaluation, factors such as footprint, turnaround time, and power usage were taken into account.

Keywords: Metastability, FPGA, Phase Shift, Auto Tuning, Flip Flops, Carry Chain, Post-Processing, Verilog HDL, QuartusII.

1. INTRODUCTION

Random number generators are an essential component of today's security systems. True random number generators (TRNGs) are the only cryptographic primitives that can create truly unexpected bits for the purpose of creating secrets in symmetric and public-key cryptography due to their physical unclonable functions (PUFs). Furthermore, several precautions are utilized to avoid SCA on cryptographic systems, many of which rely on random number generators. A high area penalty is occasionally used to prevent low-latency side-channel assaults (SCAs). By reducing the amount of space accessible for random number generator (RNG) operations, this approach decreases exposure to SCAs. Furthermore, a TRNG may not always be capable of providing enough random bits for these methods inside a single processing cycle. As a result, pseudo-random number generators (PRNGs) are utilized since they can generate random masks faster. A cryptographically safe pseudorandom number generator (PRNG) generates statistically perfect random bits. However, because it is deterministic, its output can be predicted with absolute confidence if its internal state is known or can be inferred. Professionals in information security regularly use terms such as "128-bit AES encryption ensures security" and "2048-bit authentication provides protection." People routinely express worries about the security solutions they employ, namely the cryptographic technologies that support them. The AES, RSA, and ECC cryptographic algorithms are well-known for their high level of security, which makes them difficult to crack. Despite the critical role that random number generators play in securing our identities and keeping our data private, secure, and always available, there have been surprisingly few public remarks about the dependability of these security system components. System designers usually prioritize power consumption and bit generation rate over bit unpredictability.

2. LITERATURE SURVEY

The suggested hardware solution includes a true random number generator and generates random numbers using resilient chaos and switched capacitors.

The method described in this study lowers the impact of power supply voltage on the generated random bit stream. The proposed True Random Number Generator (TRNG) can generate bitstreams at a maximum rate of 60 kbit/s, and its integrity has been confirmed.

Field-programmable gate arrays (FPGAs) generate truly random values by utilizing circuit metastability and adaptive feedback management.

A system that uses a feedback loop to track the probabilities of output bits is studied and scored in this study. If a probability bias is discovered, the system adjusts the delay using Programmable Delay Lines (PDLs) to return to its metastable working zone.

TRNGs Get a Slick New Look The use of a Latched Ring Oscillator is being investigated in order to ensure compatibility with Field-Programmable Gate Arrays (FPGAs).

The performance of the TRNG has been validated using variations in supply voltage and temperature. The design was successfully implemented using Xilinx Spartan-6 devices, according to the linked study. The measured throughput at 50 MHz is 0.76 Mbit/s, with an expected entropy per bit of about 7.99834.

The current study focuses on the integration of Time-to-Digital Conversion with a Three-Edge Ring Oscillator. The TRNG is the best choice for random number generation due to its minimal resource utilization and simple architecture, high throughput, and big minimum entropy rate. A total of 33 segments are used for the TRNG's digital noise source and post-processing, resulting in a throughput of 12.5 Mbps.

3. SCOPE AND AIM OF THE PROJECT

Scope of the project

TRNGs based on FPGAs have found broad use because unbiased and dependable random numbers are required in many fields of computing, including encryption, simulation, and scientific computing.

Aim of the project

The procedures are carried out at a very fast rate in order to provide a constant stream of random numbers suited for use in high-speed applications. To generate random numbers, a clock manager can be utilized. The FPGA-based design's flexibility and adaptability were considered during the examination of the generator's area, latency, and power consumption.

4. PROPOSED METHOD

The system clock, also known as the digital clock management (DCM) input clock clk_in, is what drives the clock input of a flip-flop (FF). The DCM generates the output clock, clk_out, at the same power frequency as the flip-flops' data input. When Clk_in and Clk_out arrive at the FF's inputs at the same time, the setup/hold time constraints of the flip-flop (FF) are breached, driving the FF into the metastable state. The suggested design uses the DPS attribute to force one or more flip-flops (FFs) into the metastable area. To compensate for the disparity in routing latencies, DCM phase shift is utilized. We use a unique hardware implementation of the clock control basic to quickly develop a True Random Number Generator (TRNG). Instead of adjusting the rate at which the clocks are sent out,

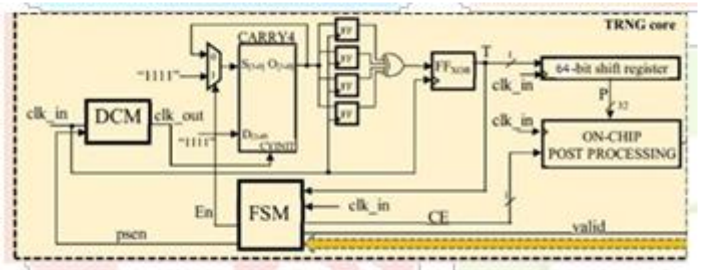


Fig.1: TRNG Core

The suggested design uses dynamic phase shifting (DPS) to induce metastability in one or more flip-flops (FFs). Once this criterion is reached, the suggested technique may automatically change the phase shifting of the DCM, allowing for the rapid start of random sequence production.

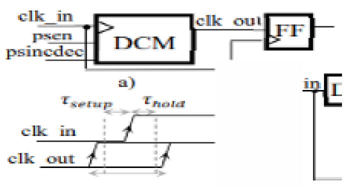


Fig.2: T-shift

We also present a novel way for boosting the unexpected factor by combining the FPGA slice's carry-chain primitive with a programmable feedback system.

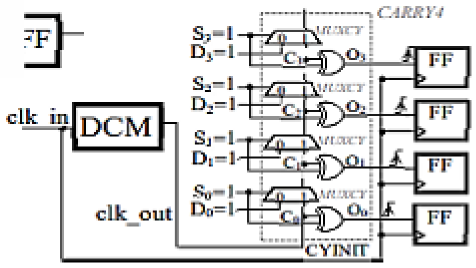


Fig.3: Carry4

The total of the outputs of a CARRY4 chain (O[3:0]) serves as the input to four flip-flops. The clk_out signal controls the CYINIT input of the CARRY4 component. The propagation delay created by the multiplexer, indicated by r, which occurs between the output positions i and i+1, causes the generic signal Oi+1 to be delayed with respect to Oi. In this situation, i can be any positive number between 0 and 2. Because the r-shift is bigger than the T mux, the data inputs to the flip-flop will suffer more precise phase shifting if the T-shift is large enough. Because all four FFs (feedback functions) sample the same constant value, the FFXOR random number generator's output T is always 0.

The feedback signals O[3:0] control the selectors S[3:0] of the multiplexers in the carry chain. When En is asserted to logic high (1), signal S[3:0] bears the binary value "1111", indicating that the auto-calibration phase is still running. Signal T is routed into a Finite State Machine (FSM), which uses feedback to manage the Digital Clock Manager's (DCM) setup signals.

The suggested technique is intended to cause a race condition at the CARRY4 network's XOR gates. According to empirical findings, at least one of the four FFs reaches the metastable area after the auto-calibration process in many placement locations. This self-calibration method takes roughly 160 ticks of the clock on average.

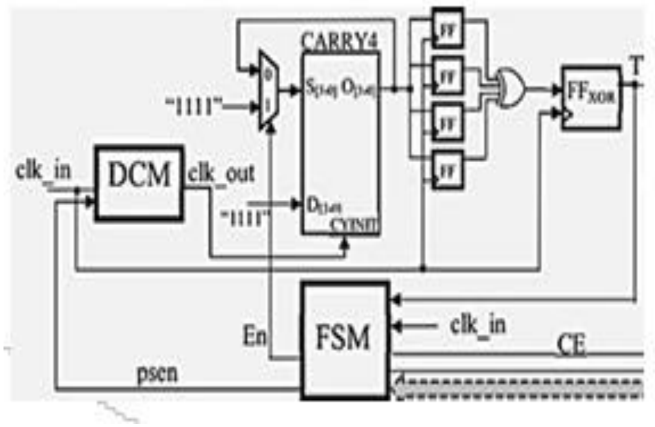


Fig.4: Auto calibration & Feedback Loop

A 10Mb sequence created by an XOR gate has a very close to 50% distribution of 0s and 1s. In the suggested method, the signal T represents the raw random bit, which is created at a rate equal to the system clock frequency. The DCM is utilized here as an additional strategy to increase the degree of unpredictability presented by the signal T. Finally, a concise summary

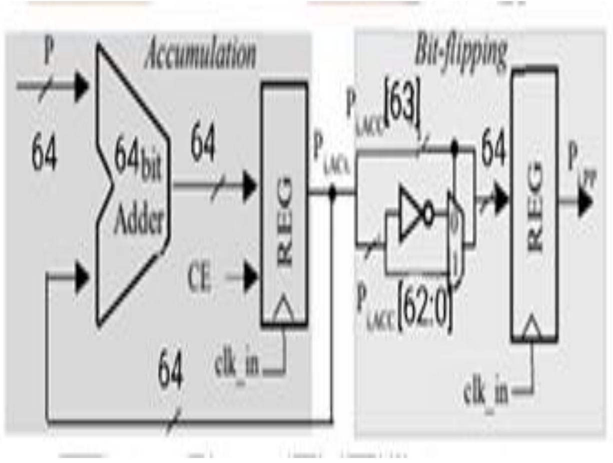


Fig.5: On chip Post-Processing

To avoid impeding the generation of new bits, it is best to use a single DSP block for on-chip post-processing.

5. SOFTWARE USED

QUARTUS II

The Altera Quartus II design environment is versatile and extensive, and it can be swiftly tailored to unique project requirements. You can create a system-on-a-chip (SOPC) within this all-encompassing environment. The Quartus II software is an all-in-one design tool for FPGAs and CPLDs that covers every phase of the process. **All of the following are required:** project completion, design creation, technology landscape analysis, simulation implementation, waveform editor use, and simulator tool implementation.

VERILOG HDL

HDL is an abbreviation for "Hardware Description Language" in this context. Any digital system is represented by a Register Transfer Level (RTL), and this RTL is defined by Hardware Description Languages (HDLs). The goal of this study is to provide a full description of how data processing and transfer occur within the framework of design.

This field's primary operations include function/behavior design, logic/circuit design, physical design, layout verification, fabrication, and testing.

The modulus concept: Here's how the syntax works:

6. RESULTS

Throughout the random number generation process, High Speed Area Power Dissipation is taken into account.

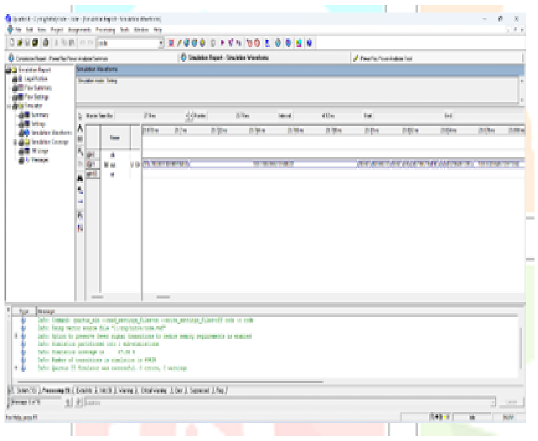
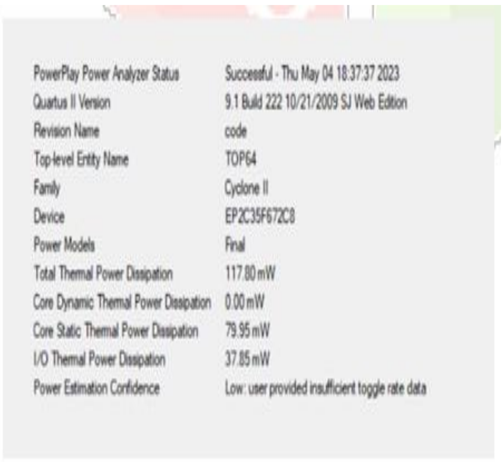
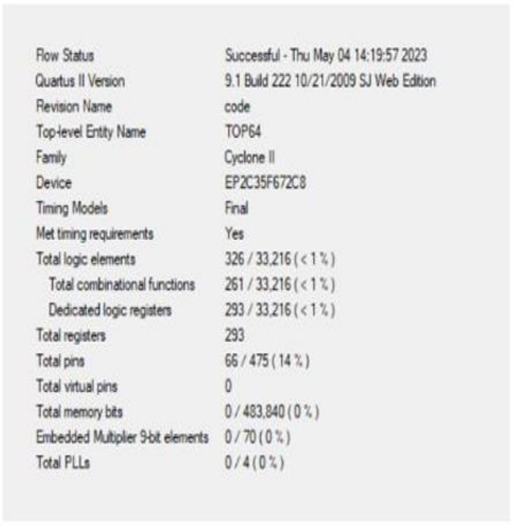


Fig.6: Simulation Results



PowerPlay Power Analyzer Status	Successful - Thu May 04 18:37:37 2023
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	code
Top-level Entity Name	TOP64
Family	Cyclone II
Device	EP2C35F672C8
Power Models	Final
Total Thermal Power Dissipation	117.80 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	79.95 mW
I/O Thermal Power Dissipation	37.85 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Fig.7: Power Dissipation



Flow Status	Successful - Thu May 04 14:19:57 2023
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	code
Top-level Entity Name	TOP64
Family	Cyclone II
Device	EP2C35F672C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	326 / 33,216 (< 1 %)
Total combinational functions	261 / 33,216 (< 1 %)
Dedicated logic registers	293 / 33,216 (< 1 %)
Total registers	293
Total pins	66 / 475 (14 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Fig.8: Area Cnsumption

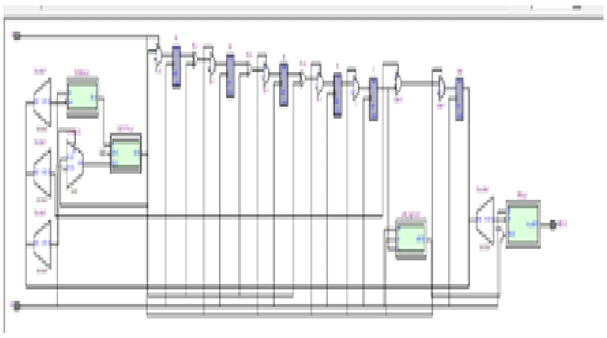


Fig.9: RTL Schematic

7. ADVANTAGES AND APPLICATIONS

ADVANTAGES:

Randomization algorithms have been improved, and tougher safety protocols have been implemented.

APPLICATIONS:

There are further applications for cost-effective authentication, key generation, and one-time passwords (OTPs).

8. CONCLUSION

A new design utilizing DCM-based TRNG has been designed to aid in the smooth inclusion of FPGA devices. DCM's hardware primitives are used to change the phase difference between two clock signals dynamically. Following these signals, metastability is activated, which promotes unpredictability. A basic finite state machine (FSM) computes the requisite phase difference automatically. A new usage of the CARRY4 hardware feature can be made to boost the randomness of the generated bits. In addition, in order to reduce delays, a method for executing postprocessing directly on the chip is demonstrated here.

REFERENCES

1. M. Drutarovsky, and P. Galajda, "A Robust Chaos-Based True Random Number Generator Embedded in Reconfigurable Switched-Capacitor Hardware," in Proc. of 17th International Conference Radioelektronika, pp. 1-6, Apr. 2007.
2. M. Majzoobi, F. Koushanfar, and S. Devadas, “FPGA-based true random number generation using circuit metastability with adaptive feedback control,” in Proc. Crypt. Hard. Embedded Syst.(CHES), 2011, pp. 17–32.
3. H. Hata, and S. Ichikawa, “FPGA Implementation of Metastability-Based True Random Number Generator,” IEICE Trans. Inf. & Syst., vol.E95-D, no. 2, pp. 426-436, Feb 2012.
4. R. Della Sala, D. Bellizia and G. Scotti, "A Novel Ultra- Compact FPGACompatible TRNG Architecture Exploiting Latched Ring Oscillators," in IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 1672-1676, March 2022.
5. N. N. Anandakumar, S. K. Sanadhya, and M. S. Hasmi, “FPGA-based True Random Number Generation Using Programmable Delays in Oscillatorrings,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 3, pp. 570- 574, March 2020.
6. H. Martin, P. Peris-Lopez, J. E. Tapiador, and E. San Millan, “A New TRNG Based on Coherent Sampling With Self-Timed Rings,” IEEE Trans.